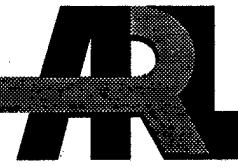


ARMY RESEARCH LABORATORY



Contract Req  
Software Requirements Analysis  
Version 1: A C-BASS Component

by Denis McGurin

ARL-MR-458

August 1999

19990819 088

Approved for public release; distribution is unlimited.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

# **Army Research Laboratory**

Aberdeen Proving Ground, MD 21005-5067

---

---

**ARL-MR-458**

**August 1999**

## **Contract Req Software Requirements Analysis Version 1: A C-BASS Component**

**Denis McGurin**

Corporate Information and Computing Directorate, ARL

---

---

Approved for public release; distribution is unlimited.

---

---

## **Abstract**

---

This document contains the software requirements analysis for a prototype of Contract Request Version 1.0 (Contract Req). As a component of the Corporate Business Application Software System (C-BASS), this application automates the preparation of the requester's Procurement Data Package (PDP). The document follows the process of structured analysis, or step-wise refinement of requirements, as applied to the development of Contract Req. The "environmental model" includes a high-level system description, followed by a context diagram and a list of events to which the system must respond. The "behavioral model" includes a data flow diagram (DFD) for each of the five Contract Req subsystems. From this representation, the basic functional specifications are derived and represented in structured English (or program design language). The final segment of the document includes a data dictionary.

# Table of Contents

	<u>Page</u>
<b>List of Figures .....</b>	v
<b>List of Tables .....</b>	v
<b>1. Introduction .....</b>	1
1.1 Contract Req Prototype .....	1
1.2 Development Plan and Project Schedule .....	1
1.3 Contents of the Report.....	1
<b>2. Structured Analysis Overview .....</b>	2
<b>3. System Overview .....</b>	3
3.1 Purpose and Required Functionality .....	3
3.2 Event List .....	5
<b>4. System Requirements.....</b>	5
<b>5. Functional Specifications.....</b>	9
5.1 Contract Req Subsystems.....	10
5.2 Subsystems Data Flow Diagrams.....	11
5.3 Processing Narration .....	14
<b>6. Data Dictionary .....</b>	22
<b>7. References .....</b>	33
<b>Distribution List .....</b>	35
<b>Report Documentation Page.....</b>	37

**INTENTIONALLY LEFT BLANK.**

## List of Figures

<u>Figure</u>		<u>Page</u>
1.	Context Diagram for Contract Req .....	4
2.	Major Subsystems of Contract Req.....	10
3.	Prepare Contract Request Process.....	12
4.	Approve Funds Process .....	12
5.	Approvals Process .....	13
6.	Edit Contract Request Process .....	13
7.	Inquiries Process.....	14

## List of Tables

<u>Table</u>		<u>Page</u>
1.	High-Level Requirements for the Contract Req Prototype .....	4

**INTENTIONALLY LEFT BLANK.**

# **1. Introduction**

Contract Request Vision 1.0 is a component of the Corporate Business Application Software System (C-BASS) cluster of applications, an integrated set of Lotus Notes and Web-based software to support U.S. Army Research Laboratory (ARL) electronic workflow and task automation. (Throughout this report, Contract Request Version 1.0 is abbreviated as Contract Req.) The motivating force behind this work effort has been ARL downsizing and the findings contained in the Business Process Reengineering (BPR) report on the contracting process published in 1995 [1]. The BPR “To-Be Model: Formal Contracts” [1] identified potential process improvements — some of which require computer automation — that would increase productivity of contracting operations at ARL. Development of a full-production Contract Req will proceed in phases, using an incremental, evolutionary approach. The system described in this document is the first version in this process.

**1.1 Contract Req Prototype.** The purpose of the Contract Req prototype is to provide a secure client/server system that provides for the processing of contract requests. This proof-of-principle prototype will alleviate some of the risks involved in implementing new technologies used to build the ARL Intranet. The project will also refine requirements described in the ARL BPR To-Be Model [1].

**1.2 Development Plan and Project Schedule.** The “Contract Req Software Development Plan” [2] states a definition of the problem; gives an overview of technical, management, and reliability issues; and provides a detailed project schedule. The report given herein covers the work accomplished to solidify user requirements (primarily drawn from existing high-level design documents) and the analytical expansions used to derive a data flow model, a pseudocode representation of processing, and a data dictionary.

**1.3 Contents of the Report.** This document presents the results of a structured analysis used to derive the software requirements for Contract Req, starting with the baseline given in the BPR “To-Be Model: Formal Contracts” [1]. The body of the report contains five sections:

- “Structured Analysis Overview” - briefly explains the methodology used to extract the functional specifications.
- “System Overview” - delineates the basic Contract Req concept and outlines the high-level requirements.
- “System Requirements” - breaks the more generic statements into low-level, derived requirements and describes each in detail.
- “Functional Specifications” - discusses the products of the structured analysis (i.e., the data flow diagrams and the structured English narrative) for each subsystem of Contract Req.
- “Data Dictionary” - lists each of the Contract Req data elements, giving a full description and type for the data model.

## **2. Structured Analysis Overview**

Modern software engineering utilizes structured analysis [3] as a powerful methodology for developing system specifications. Through a series of step-wise refinements, detailed delineation of the system’s components and their behavior are extracted from high-level descriptions of system features and functions. In other words, primary system elements are broken down into progressively more detailed levels of processes, and the data paths between these processes are defined. Three modeling tools facilitate this decomposition: (1) data flow diagrams (DFDs), (2) structured English process narratives (either pseudocode or program design language [PDL]), and (3) a data dictionary. Accuracy and precision in progressively expanding design definitions are critical to successful system development.

The results of this analytical approach are systematic elaborations of product requirements, typically expressed as parts of two separate models:

- An environmental model that defines the system's interfaces to the outside world (see section 3, “System Overview”).
- A behavioral model that defines the internal behavior the system must exhibit in order to deal with the environment (see section 4, “System Requirements,” and section 5, “Functional Specifications”).

### 3. System Overview

The environmental model typically consists of three components: (1) a concise statement of the system’s purpose or required functionality, (2) a context diagram, and (3) an event list. The context diagram is the highest level DFD. It shows the system as a single process, including user interaction and communication with external systems, as well as data flow input and output. The event list provides an index of outside stimuli to which the system responds.

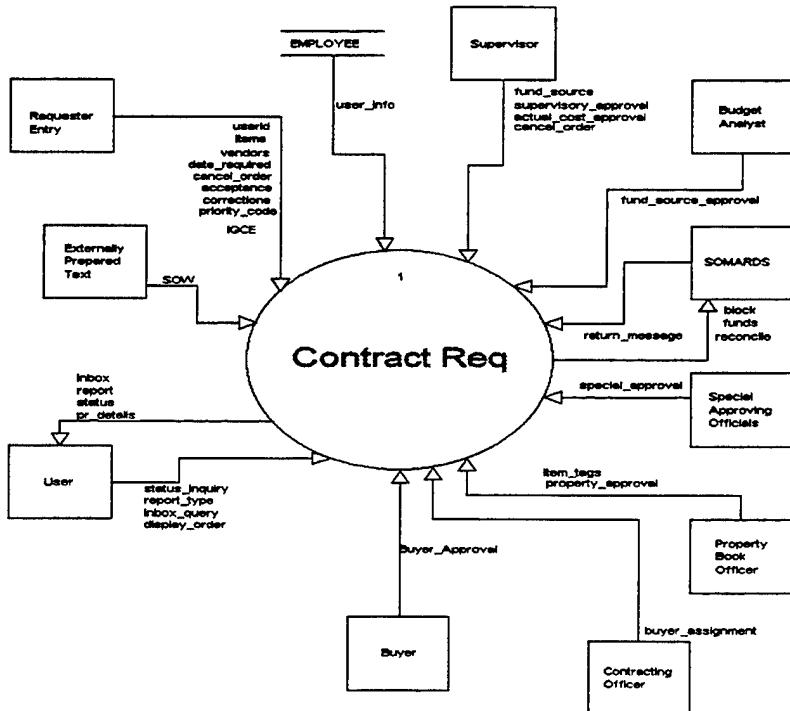
**3.1 Purpose and Required Functionality.** As a system, the Contract Req prototype provides a secure, automated means for the preparation, submission, approval, and tracking of certain aspects of the contracting process. The complete contract life cycle is complex and multistaged. Contract Req currently automates only the preparation of the requester’s Procurement Data Package (PDP). See “Contract Req Software Development Plan” [2] for an explanation of the exact portions of the contract process automated by this product.

Table 1 lists the high-level requirements for the Contract Req prototype and gives a general description of what the requirement involves.

Figure 1 shows the context diagram for the prototype. This diagram displays the external entities (e.g., users, functional areas, and legacy systems), represented by the squares, with which Contract Req communicates. The data flowing into and out of Contract Req are represented by the arrows. A few elements on the DFD need additional explanation. First, the external entity “User” represents all users of the system. The data flow associated with this box is limited to

**Table 1. High-Level Requirements for the Contract Req Prototype**

Requirement	General Description
Security	Provide security measures to prevent unauthorized access to the system and its data, and keep authorized users from performing tasks not allowed in their roles.
Contract request preparation	Provide a means for the requesters and functional users to input/edit relevant information pertaining to a contract request.
Automated request routing	Automate the process of routing contract requests to the various functional areas.
Electronic approval	Provide a means for approving officials and functional users to electronically approve or reject a contract request.
Request tracking	Allow users to track the status of active contract requests currently in the system.
Legacy system interface	Implement automated interfaces to Standard Operations and Maintenance Army Research Development System (SOMARDS).
Reporting	Provide users and management with a means for reporting cycle times and costs.



**Figure 1. Context Diagram for Contract Req.**

display of information. All the other external entities (e.g., “Requester,” “Supervisor,” etc.), and their corresponding data flows, show the specific information that is passed to Contract Req, either by that user or by the system. The data store EMPLOYEE contains user information such as name, phone number, address, office symbol, etc.

### **3.2 Event List.** The following are the events to which the system must respond.

- Requester initiates new contract request.
- Requester prepares request.
- Requester corrects rejected request.
- Requester, supervisor, or budget analyst completes fund source.
- Requester or supervisor cancels request.
- Budget analyst certifies fund source.
- SOMARDS certifies and commits funds.
- Special approving official(s) approves request item(s).
- Property book officer approves request.
- Contracting officer assigns buyer to request.
- Supervisor approves actual costs.
- User submit status inquiry.
- User request report.

## **4. System Requirements**

Antecedent studies and legacy systems also contribute to Contract Req’s requirements. The “To-Be Model: Formal Contracts” [1] was produced during the BPR study in 1995. However, for some areas, this document lacks detail, making it necessary to derive missing elements. Additionally, the limited scope of a prototype necessitates leaving out some of the more complicated or vague automation requirements of the “To-Be Model” or where the requirement already exists in a Department of Defense (DOD) standard system (e.g., functionalities handled by the Standard Army Automated Contract System [SAACONS]). The “Contract Req Software

Development Plan” [2] more fully addresses the constraints on the prototype and the requirements of the legacy systems.

The following elaboration more fully defines the proscribed system tasks enumerated in Table 1.

## E1

### E2 Security

#### E21 Prevent unauthorized access

*Description* — Prevent unauthorized access to the system and its data.

*Source* — Derived, due to the nature of the system.

*Interfaces to major functions and external entities:*

User

#### E22 Enforce role restrictions

*Description* — Prevent users from performing tasks or accessing/editing data that are out of the scope of their role.

*Source* — BPR “To-Be Model” document, requirement A151.

*Interfaces to major functions and external entities:*

User

Approvals

Edits

Employee address book (for roles).

## E3 Contract request preparation

### E31 Create new contract request

*Description* — Allow the requester to create a new contract request with preliminary user information filled in.

*Source* — BPR “To-Be Model” document, Automation Requirements section, requirement A151

*Interfaces to major functions and external entities:*

User  
Security  
Employee address book (for user info)

**E32 Select items**

*Description* — Provide a means for the requester to enter item descriptions, specifications, quantities, and estimated costs.

*Source* — BPR “To-Be Model” document, “Automation Requirements” section, requirement A151.

*Interfaces to major functions and external entities:*

User  
Security

**E33 Complete contract request**

*Description* — Provide a means for the requester and/or approving supervisor to complete the contract request.

*Source* — BPR “To-Be Model” document, Automation Requirements section, requirements A151.

*Interfaces to major functions and external entities:*

User  
Security

**E34 Item tag input**

*Description* — Provide a means for the Property Book Officer to enter item tags.

*Source* — BPR “To-Be Model” document, Automation Requirements section, requirements A151.

*Interfaces to major functions and external entities:*

User  
Security

**E35 Edit contract request**

*Description* — Provide a means for users to edit certain request details as needed.

*Source* — Derived, due to the need for making corrections to rejected contract request.

*Interfaces to major functions and external entities:*

User  
Security

#### E4

#### E5 Automated routing

*Description* — Automate the process of routing contract requests to the various functional areas and approving officials.

*Source* — BPR “To-Be Model” document, Automation Requirements section, requirements A151.

*Interfaces to major functions and external entities:*

Security  
Employee address book (for default routing)

#### E6 Electronic approval

*Description* — Provide a means for approving officials and functional users to approve or reject a contract request.

*Source* — BPR “To-Be Model” document, Automation Requirements section, requirements A151.

*Interfaces to major functions and external entities:*

User  
Security

#### E7 Request tracking

*Description* — Allow users to track the status of active contract requests currently in the system.

*Source* — User requested.

*Interfaces to major functions and external entities:*

User

## **E8      Legacy system interfaces**

### **E81    Budget legacy system interface**

*Description* — Provide an electronic interface to the legacy financial system (SOMARDS) that automates the certification and commitment of funds.

*Source* — “To-Be Model” section, process model diagram A12, process A153.

## **E9**

## **E10    Reporting**

*Description* — Provide users and management with a means for reporting cycle times and costs.

*Source* — Derived from ARL corporate experience. Not specifically noted in BPR Formal Contracts documentation.

*Interfaces to major functions and external entities:*

User

Security

## **E11    Navigation**

*Description* — Provide users with a means for navigating to the various functional areas within the system.

*Source* — Derived from the requirements listed previously.

*Interfaces to major functions and external entities:*

User

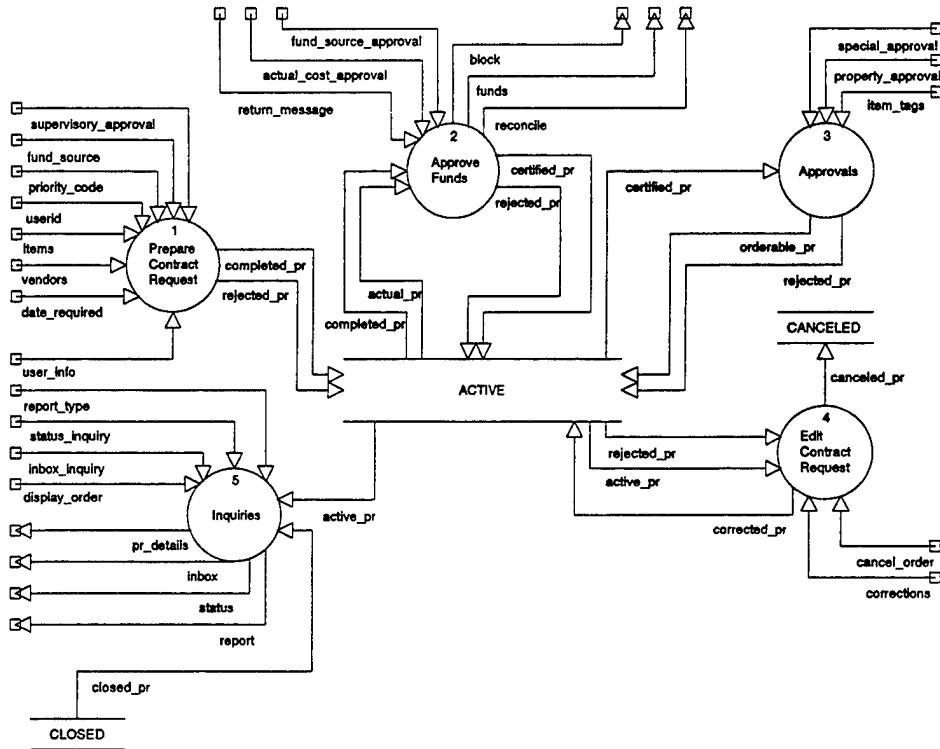
## **5. Functional Specifications**

The behavioral model expands the results from the environmental model to more fully define how the system performs prescribed tasks. Typical representations in this mode are (1) concise flow charts showing how information is transformed as it moves through the system and subsystems; (2) a set of structured English statements forming a processing narration based on

data types, control structures, and transformations; and (3) a data dictionary defining each data item.

**5.1 Contract Req Subsystems.** The eight functions listed in the previous section identify the major components of Contract Req: (1) security, (2) contract request preparation, (3) automated routing, (4) electronic approval, (5) request tracking, (6) legacy system interfaces, (7) reporting, and (8) navigation. These categories consolidate into five subsystems: (1) prepare contract request, (2) approve funds, (3) obtain approvals, (4) edit contract requests, and (5) inquire about status.

Figure 2 is a DFD showing the major functional subsystems of Contract Req. Each of the five bubbles in the diagram represents a major subsystem or process of Contract Req, with the arrows showing the data flowing into and out of the processes.



**Figure 2. Major Subsystems of Contract Req.**

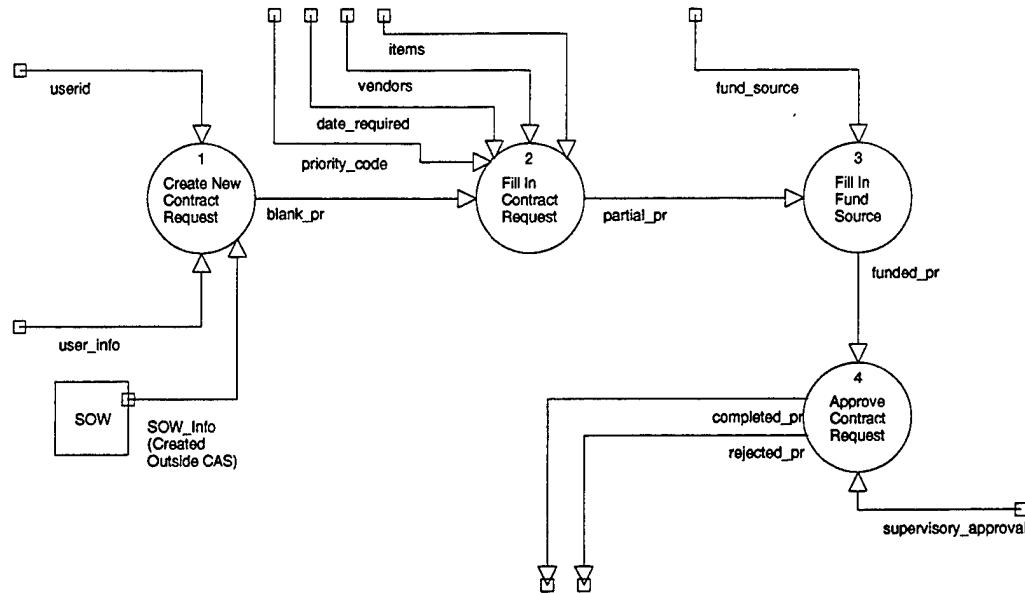
The data store ACTIVE—a Lotus Notes database located in the center of the diagram—holds all the active contract requests, each waiting for the various users to perform their functions on them. The CLOSED and CANCELED data stores contain contract requests that have been closed out and canceled, respectively. The small squares along the outer edges of this DFD are interfaces to the outside world.

No process bubble for security appears at this level because the Lotus Notes application environment handles security and verifies user authorization. Additionally, enforcement of role restrictions are handled within each subsystems, as detailed in the following sections.

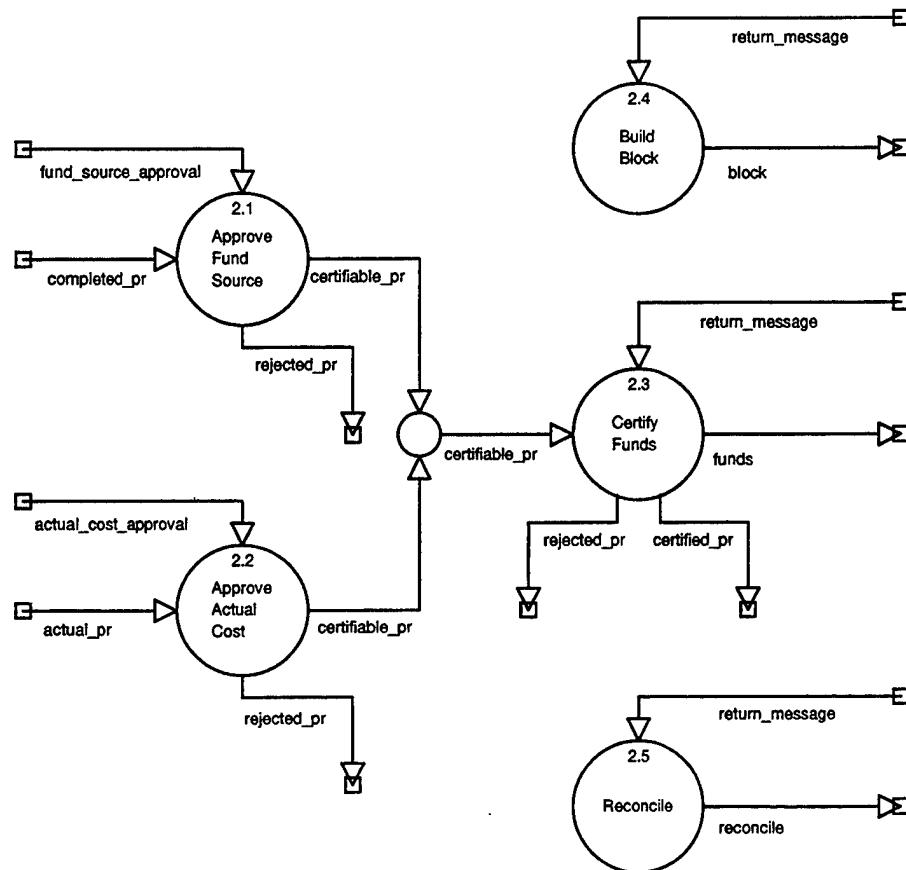
**5.2 Subsystems Data Flow Diagrams.** System objects and operations can be coherently represented as DFDs. A DFD can be used to capture system concepts and components at any level of abstraction. Each of the following five DFDs (Figures 3–7) provides more detail for the information flow and functionality of each of the identified Contract Req subsystems.

Figure 3 shows the DFD for the “Prepare Contract Request” subsystem. The major inputs to this process (and its basic functions) are the requester information (derived from the user supplied userid and the EMPLOYEE data store), item details, vendor information (if known at this time), and the date by which the items are required. The fund source completes the information for the request, and the supervisory approval puts the request into the contracting cycle.

Figure 4 shows the DFD for the “Approve Funds” subsystem. This subsystem, besides having interfaces to users for approvals, also has connections to the SOMARDS legacy system. The “Build Block” process is executed at the start of the day and creates the transaction block that will be used by Contract Req for the remainder of the day. As eligible contract requests are created during the course of the day, the “Certify Funds” process queries SOMARDS and grabs the returning message. Depending on the results of this query, the request is either certified or rejected (with explanation). At the end of the day, the “Reconcile” process is executed to balance the transaction block.

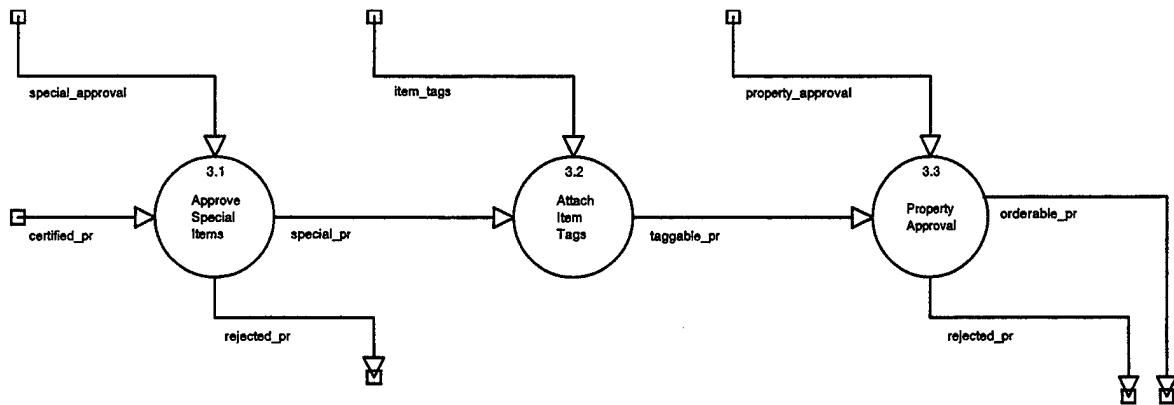


**Figure 3. Prepare Contract Request Process.**



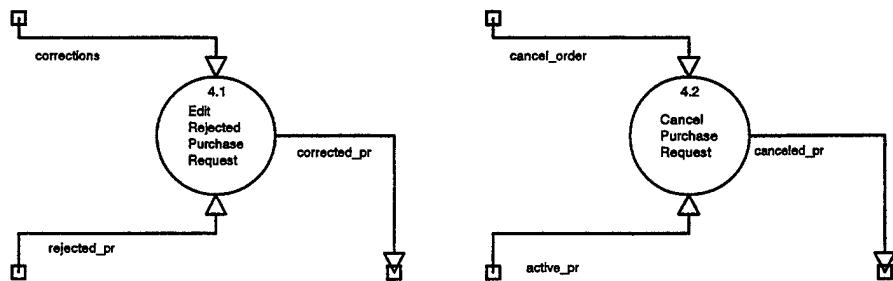
**Figure 4. Approve Funds Process.**

The “Approvals” process is shown in Figure 5. At this point, the various approving officials attach their approval or rejection to the request. The property book officer also attaches item tags to the individual items in order to flag them during receipt of the shipment.



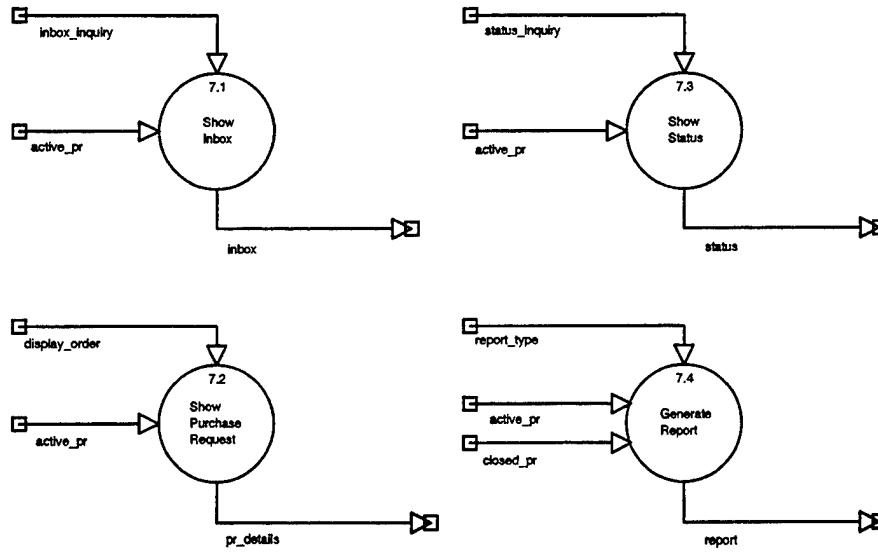
**Figure 5. Approvals Process.**

Figure 6 diagrams the “Edit Contract Request” process. The rejected contract request is displayed for the requester, who then enters the corrections. What fields within the request the user can edit depends on where the rejection came from and how far along in the approval process the request has traveled.



**Figure 6. Edit Contract Request Process.**

The “Inquiries” process is diagrammed in Figure 7. The processes shown in this figure are used to display to the user (1) pending actions (inbox), (2) contract request details, (3) the status of requests, and reports.



**Figure 7. Inquiries Process.**

**5.3 Processing Narration.** Having captured the flow of information and identified data objects, each transformation can be further expanded by using the stylized notation of structured English. Basic procedural constructs are combined with English phrases to give a concise description for each major operation listed in the prescribed tasks analysis given in section 4.

## E1 Security

### E11 Login

*Input* — userid, passwd

*Process*:

REPEAT

    GET from user the **userid, passwd**

    UNTIL VALID **userid, passwd**

    ALLOW login

*Output* – Access on Success, Error Message on Failure

### E12 Role

*Input* — role, action

*Process*

    GET from EMPLOYEE the role using **requester\_userid**

IF VALID **action** for role THEN

    EXECUTE **action**

ELSE

    NULL

ENDIF

*Output* — action\_results

## E2    Prepare contract request for ordering

### E21    Create new contract request

*Input* — pr, requester\_userid, user\_info

*Process 1.1*

    GET from EMPLOYEE the user\_info using

        requester\_userid

        SET in pr the requester\_userid

        SET in pr the user\_info using user\_info

*Output* — blank\_pr

### E22    Fill in contract request

*Input* — blank\_pr, priority\_code, items, vendors, date\_required

*Process 1.2*

    GET from user the priority\_code

    SET in pr the priority\_code

    DO WHILE there is another item to add

        GET from user the item

        SET in pr the item

    ENDDO

    GET from user the specifications

    SET in pr the specifications

    DO WHILE there is another vendor to add

        GET from user the vendor

        SET in pr the vendor

    ENDDO

GET from user the **date\_required**

SET in pr the **date\_required**

*Output* — partial\_pr

#### E23 Fill in fund source

*Input* — partial\_pr, fund\_source

*Process 1.3*

GET from user the **fund\_source**

SET in pr the **fund\_source**

*Output* — funded\_pr

#### E24 Correct contract request

*Input* — rejected\_pr, corrections

*Process 4.1*

DISPLAY to user **rejected\_pr** and explanation

DO WHILE there are more **corrections**

    GET from user the **corrections**

    SET in pr the **corrections**

ENDDO

*Output* — corrected\_pr

#### E25 Cancel request

*Input* — active\_pr, cancel\_order

*Process 4.2*

DISPLAY to user the **active\_pr**

GET from user the **cancel\_order**

PUT **canceled\_pr** into CANCELED

*Output* — cancelled\_pr

### E3 Routing

#### E31 Automated routing

*Input* — active\_pr

*Process* — TBD

*Output* — active\_pr

### **E32 Manual routing**

*Input* — active\_pr

*Process* — TBD

*Output* — active\_pr

### **E33 Assign Buyer**

*Input* — orderable\_pr

*Process 5.1*

DISPLAY to user the **orderable\_pr**

GET from user the **buyer\_assignment**

SET in **pr** the **buyer\_assignment**

SET in **pr** the **inbox\_location** to buyer

*Output* — assigned\_pr

## **E4 Approvals**

### **E41 Supervisory approval**

*Input* — funded\_pr, supervisory\_approval

*Process 1.4*

DISPLAY to user the **funded\_pr**

GET from user the **supervisory\_approval**

IF **supervisory\_approval** is ‘Yes’ THEN

    SET in **pr** the **supervisory\_approval** to ‘Yes’

    SET in **pr** the **request\_date** to today’s date

    SET in **pr** the **inbox\_location** to budget

ELSE

    GET from user the **explanation**

    SET in **pr** the **supervisory\_approval** to ‘No’

    SET in **pr** the **explanation**

    SET in **pr** the **inbox\_location** to requester

ENDIF

*Output* — completed\_pr, rejected\_pr

## **E42 Fund source approval**

*Input* — completed\_pr, fund\_source\_approval

*Process 2.1*

```
DISPLAY to user the completed_pr
GET from user the fund_source_approval
IF fund_source_approval is 'Yes' THEN
    SET in pr the fund_source_approval to 'Yes'
    SET in pr the inbox_location to certification
ELSE
    GET from user the explanation
    SET in pr the fund_source_approval to 'No'
    SET in pr the explanation
    SET in pr the inbox_location to requester
ENDIF
```

*Output* — certifiable\_pr, rejected\_pr

## **E43 Special approval**

*Input* — certified\_pr, special\_approval

*Process 3.1*

```
DISPLAY to user the certified_pr
GET from user the special_approval
IF special_approval is 'Yes' THEN
    SET in pr the special_approval to 'Yes'
    SET in pr the inbox_location to
        property_book_officer
ELSE
    GET from user the explanation
    SET in pr the special_approval to 'No'
    SET in pr the explanation
    SET in pr the inbox_location to requester
ENDIF
```

*Output* — special\_pr, rejected\_pr

#### **E44 Property approval**

*Input* — special\_pr, property\_approval

*Process 3.3*

```
DISPLAY to user special_pr
GET from user property_approval
IF property_approval is 'Yes' THEN
    SET in pr the property_approval to 'Yes'
    SET inbox_location to contracting_officer
ELSE
    GET from user explanation
    SET in pr the property_approval to 'No'
    SET in pr the explanation
    SET in pr the inbox_location to requester
ENDIF
```

*Output* — orderable\_pr, rejected\_pr

#### **E45 Actual cost approval**

*Input* — actual\_pr, actual\_cost\_approval, explanation

*Process 2.2*

```
DISPLAY to user actual_pr
GET from user actual_cost_approval
IF actual_cost_approval is 'Yes' THEN
    SET in pr the actual_cost_approval to 'Yes'
    SET in pr the inbox_location to buyer
ELSE
    GET from user explanation
    SET in pr the actual_cost_approval to 'No'
    SET in pr the explanation
    SET in pr the inbox_location to requester
ENDIF
```

*Output* — certifiable\_pr, rejected\_pr

## E46 Buyer approval

*Input* — assigned\_pr, buyer\_approval, explanation

*Process 2.2*

DISPLAY to user assigned\_pr

GET from user the buyer\_approval

IF buyer\_approval is ‘Yes’ THEN

SET in pr the buyer\_approval to ‘Yes’

ELSE

GET from user explanation

SET in pr the buyer\_approval to ‘No’

SET in pr the explanation

SET in pr the inbox\_location to requester

ENDIF

*Output* — approved\_pr, rejected\_pr

## E5 Interface with legacy systems

### E51 Build block

*Input* — N/A

*Process 2.3*

PUT to SOMARDS the block

GET from SOMARDS the return\_message

*Output* — N/A

### E52 Certify funds

*Input* — certifiable\_pr

*Process 2.4*

GET from certifiable\_pr the funds

PUT to SOMARDS the block, funds

GET from SOMARDS the return\_message

IF return\_message is ‘OK’ THEN

SET in pr the certification to ‘Yes’

SET in **pr** the **inbox\_location** to **special\_approval**  
ELSE  
    SET in **pr** the **to certification** to ‘No’  
    SET in **pr** the **explanation** to **return\_message**  
    SET in **pr** the **inbox\_location** to **requester**  
ENDIF

*Output* — certified\_pr, rejected\_pr

#### E53 Reconcile

*Input* — N/A

*Process 2.5*

PUT to **SOMARDS** the **block, reconcile**  
GET from **SOMARDS** the **return\_message**

*Output* — N/A

#### E54 Upload to SAACONS

*Input* — assigned\_pr

*Process 5.1*

GET from **assigned\_pr** the **upload**  
PUT to **SAACONS** the **upload**

*Output* — N/A

#### E6 Status inquires

*Input* — active\_pr

*Process 7.3*

GET current **status** from **ACTIVE**  
DISPLAY to user the **status**

*Output* — status

#### E7 Generate reports

*Input* — report\_type

*Process 7.4 — TBD*

*Output* — report

## E8 Navigation

### E81 Navigate

*Input — TBD*

*Process — TBD*

*Output — TBD*

### E82 Logout

*Input — N/A*

*Process — TBD*

*Output — N/A*

## 6. Data Dictionary

While DFDs and pseudocode (structured English) are important to system specifications, additional information is required for a complete analytical model. The content of each data or control item should be more fully identified. A data dictionary is a quasi-formalism for describing content of information as it flows through the system. The standard notation conventions are

<u>Notation</u>	<u>Meaning</u>
=	is composed of
+	and
[ ]	either - or
{ } <sup>n</sup>	n repetitions of
( )	optional data
* *	comments

Contract Req prototype's data dictionary appears in following text. Each left-hand element is taken from the DFD and the PDL representations of the system. These data items are then given an expanded, unambiguous definition in the right-hand column.

acceptance = \*requester accepts or returns shipment item\*  
[“Yes” | “No”]

ACTIVE = {active\_pr}

action = \*\*  
**TBD**

action\_results = \*\*  
**TBD**

active\_pr = \*purchase request at some point in the approval cycle\*

actual\_cost = \*actual cost of an item\*  
\*units: dollars\*

actual\_cost\_approval = \*supervisory approval of the total actual cost of the request\*  
[“Yes” | “No”]

actual\_pr = \*purchase request where the total actual cost is greater than the total estimated cost\*  
approved\_pr + {actual\_cost} + {item\_tot\_act\_cost} + total\_actual\_cost

address = \*\*  
street\_address + (mail\_stop) + city + state + postal\_code + (country)

approved\_pr = \*purchase request that has procurement buyer approval\*  
assigned\_pr + buyer\_approval

assigned\_pr = \*purchase request that has been assigned to a buyer by the contracting officer\*  
orderable\_pr + buyer\_assignment

bar\_code\_no = \*property book bar code number\*  
{alphanumeric}

batch\_no = \*SOMARDS batch number\*  
“Contract Req”

blank\_pr = \*purchase request with the requester and delivery info filled\*  
requester\_userid + user\_info

bldg\_no = \*building number\*  
 {alphanumeric}

blk\_no = \*SOMARDS block number\*  
 “ARL”

blk\_tkt\_dt = \*SOMARDS block ticket date\*  
 \*format: MMDDYY\*  
 date

block = \*SOMARDS build block data\*  
 trns\_cd + user\_auth\_key + cmd\_dsg + update\_code +  
 blk\_no + blk\_tkt\_dt + tot\_blk + batch\_no + tot\_batch

buyer\_approval = \*buyer approval of purchase request\*  
 [“Yes” | “No”]

buyer\_assignment = \*\*  
 buyer\_userid + saacons\_buyer\_code

buyer\_userid = \*\*  
 userid

cancel\_order = \*order to cancel purchase request\*  
 [“Yes” | “No”]

CANCELED = {canceled\_pr}

canceled\_pr = \*purchase request that has been canceled\*  
 active\_pr + cancel\_order

certifiable\_pr = \*purchase request that has fund source approval or actual  
 costs approved\*  
 [completed\_pr + fund\_source\_approval | actual\_pr +  
 actual\_cost\_approval]

certification = \*SOMARDS certification\*  
 [“Yes” | “No”]

certified\_pr = \*purchase request that has been certified by SOMARDS\*  
 certifiable\_pr + certification

city = \*\*  
 {alphabetic\_character}

CLOSED = {closed\_pr}

closed\_pr = \*purchase request that has been accepted by the requester\*  
tagged\_pr + {acceptance}

cmd\_dsg = \*SOMARDS CMD-DSG\*  
“I”

company\_address = \*\*  
address

company\_email = \*\*  
email\_address

company\_fax\_no = \*\*  
phone\_no

company\_name = \*\*  
{legal\_character}

company\_phone\_no = \*\*  
phone\_no

company\_poc = \*\*  
name

completed\_pr = \*purchase request that has been approved by the supervisor\*  
@doc\_ref\_no + funded\_pr + request\_date +  
supervisory\_approval

comt\_ref\_no = \*SOMARDS document reference number\*  
doc\_ref\_no

corrected\_pr = \*purchase request that has been corrected by the requester\*  
rejected\_pr + corrections

corrections = \*corrections to a rejected purchase request\*  
**TBD**

country = \*\*  
{alphabetic\_character}

cum\_btch\_value = \*SOMARDS cumulative batch total for the days certification\*  
\*units: dollars\*

date\_required = \*date shipment is required by\*  
date

delivery\_date = \*estimated date for delivery from vendor\*  
date

description = \*item description\*  
{legal\_character}

doc\_ref\_no = \*purchase request document reference number\*  
“W-” + **TBD**

email\_address = \*\*  
**TBD**

EMPLOYEE = {employee}

employee = \*employee information - the bare minimum should contain\*  
user)info + {roles}

eor = \*funding element of resource\*  
{alphanumeric}

estimated\_cost = \*estimated cost of an item\*  
\*units: dollars\*

explanation = \*rejection, cancelation, or return explanation\*  
{legal\_character}

finished\_pr = \*purchase request where the total actual cost is less than or equal to the total estimated cost\*  
approved\_pr + {actual\_cost} + {item\_tot\_act\_cost} + total\_actual\_cost

first\_name = \*a person's first name\*  
{alphabetic\_character}

fund\_source = \*\*  
jo\_no + eor

fund\_source\_approval = \*budget analyst approval of fund source\*  
[“Yes” | “No”]

funded\_pr = \*purchase request with a fund source\*  
partial\_pr + fund\_source

funds = \*funding information for SOMARDS certification\*  
trns\_cd + user\_auth\_key + cmd\_dsg + update\_code +  
blk\_no + blk\_tkt\_dt + batch\_no + rej\_rept\_director +  
doc\_ref\_no + jo\_no + eor + act\_amt

inbox = \*purchase requests requiring action from user\*  
{active\_pr}

inbox\_inquiry = \*\*  
**TBD**

inbox\_location = \*current purchase request location\*  
**TBD**

item = \*\*  
@line\_item\_no + description + unit\_of\_issue + qty +  
estimated\_cost + actual\_cost + item\_tag + acceptance +  
item\_tot\_est\_cost + item\_tot\_act\_cost

items = \*\*  
{item} + specifications

item\_tag = \*property book officer inputs ‘yes’ item is taggable,  
receiving overwrites with bar\_code\_no\*  
[“Yes” | bar\_code\_no]

item\_tags = {item\_tags}

item\_tot\_act\_cost = \*line item total actual cost\*  
\*units: dollars\*

item\_tot\_est\_cost = \*line item total estimated cost\*  
\*units: dollars\*

jo\_no = \*funding job number\*  
{alphanumeric}

last\_name = \*a person’s last name\*  
{alphabetic\_character}

line\_item\_no = \*line item number\*  
{numeric\_digit}

mail\_stop = \*mail stop or department\*  
{legal\_character}

name = \*\*  
first\_name + last\_name

nomenclature = \*SOMARDS certification comment field\*  
{alphanumeric}

office\_symbol = \*ARL office symbol\*  
“AMSRL-” + {alphabetic\_character} + “-” +  
{alphabetic\_character}

orderable\_pr = \*purchase request that can be ordered by procurement\*  
taggable\_pr + property\_approval

ordered\_pr = \*purchase request that has been ordered\*  
finished\_pr + delivery\_date + vendor + po\_number

partial\_pr = \*purchase request with items and vendors filled in\*  
blank\_pr + date\_required + priority\_code + items + vendors

phone\_no = \*a phone number\*  
{numeric\_digit}

po\_number = \*purchase order number\*  
**TBD**

postal\_code = \*postal/zip code\*  
{numeric\_digit}

pr\_details = \*details about the purchase request\*  
**TBD**

priority\_code = \*request priority code\*  
\*range:01 - 15, 99\*  
{numeric\_digit}

property\_approval = \*property book officer approval\*  
[“Yes” | “No”]

qty = \*quantity requested\*  
 {numeric\_digit}

receipt = \*shipment receipt\*  
 [“Yes” | “No” ]

received\_pr = \*purchase request that has been received\*  
 ordered\_pr + receipt

reconcile = \*end of the day SOMARDS reconcile info\*  
 trns\_cd + user\_auth\_key + cmd\_dsg + update\_code +  
 blk\_no + blk\_tkt\_dt + batch\_no + tot\_blk + tot\_batch +  
 ty\_act\_cd + cum\_btch\_value + variance

rejected\_pr = \*purchase request that has been rejected\*  
 active\_pr + explanation

rej\_rept\_director = \*SOMARDS REJ-REPT-DIRECTOR\*  
 “R”

report = \*\*  
**TBD**

report\_type = \*type of report to generate\*  
**TBD**

request\_date = \*date purchase request was approved by supervisor\*  
 date

requester\_userid = \*\*  
 userid

return\_message = \*message returned from SOMARDS process\*  
 [“processing complete” | “bad user\_auth\_key” | “wrong  
 update code” | “blk\_no/blk\_tkt\_dt already exists” |  
 “accounting class displayed” | “blk\_no/blk\_tkt\_dt doesn’t  
 exist” | “invalid jo\_no” | “invalid eor” | “insufficient funds”  
 | “duplicate comt\_ref\_no” | “cum\_btch\_value” | “make  
 changes” | “variance” ]

role = \*user role\*  
 {alphanumeric}

room\_no = \*room number\*  
 {alphanumeric}

sole\_source\_just = \*justification for using a single vendor\*  
{legal\_character}

special\_approval = \*approval from a special approving officials\*  
[“Yes” | “No”]

special\_pr = \*purchase request with special approvals\*  
certified\_pr + special\_approval

state = \*state or province\*  
{legal\_character}

status = \*\*  
**TBD**

status\_inquiry = \*\*  
**TBD**

street\_address = \*\*  
{legal\_character}

supervisory\_approval = \*approval from supervisor\*  
[“Yes” | “No”]

taggable\_pr = \*purchase request that has had item tags attached by  
property book officer\*  
special\_pr + item\_tags

tagged\_pr = \*purchase request that has been received and taggable items  
have been appropriately tagged\*  
received\_pr + item\_tags

total\_actual\_cost = \*the total actual cost of the purchase request\*  
\*units: dollars\*

total\_estimated\_cost = \*the total estimated cost of the purchase request\*  
\*units: dollars\*

tot\_batch = \*SOMARDS batch number\*  
\*units: dollars\*  
[“0.00” | cum\_btch\_value]

tot\_blk = \*SOMARDS total block\*  
\*units: dollars\*  
[“0.00” | cum\_btch\_value]

trns\_cd = \*SOMARDS transaction code\*  
[“003” | “004” | “310”]

ty\_act\_cd = \*SOMARDS action code\*  
“C”

unit\_of\_issue = \*\*  
*TBD*

update\_code = \*SOMARDS update code\*  
[“CM” | “NM”]

user\_auth\_key = \*SOMARDS user authorization key\*  
{alphanumeric}

user\_info = \*user information\*  
name + office\_symbol + phone\_no + bldg\_no + room\_no

userid = \*\*  
{alphanumeric}

variance = \*SOMARDS variance between tot\_batch and  
cum\_btch\_value - should be 0.00\*  
\*units: dollars\*

vendor = \*vendor information\*  
company\_name + company\_address + company\_phone\_no  
+ company\_fax\_no + company\_poc + company\_email

vendors = \*up to three suggested vendors\*  
{vendor} + sole\_source\_justification

**INTENTIONALLY LEFT BLANK.**

## **7. References**

1. Business Process Reengineering Office. "To-Be Model: Formal Contracts." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, October 1995.
2. McGurin, D. "Contract Req Software Development Plan." Enterprise Systems Division, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, to be published.
3. Yourdon, E. *Modern Structured Analysis*. Englewood Cliffs, NJ: Yourdon Press, 1989.

**INTENTIONALLY LEFT BLANK.**

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218	1	DIRECTOR US ARMY RESEARCH LAB AMSRL DD J J ROCCHIO 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	HQDA DAMO FDQ D SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460	1	DIRECTOR US ARMY RESEARCH LAB AMSRL CS AS (RECORDS MGMT) 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	OSD OUSD(A&T)/ODDDR&E(R) R J TREW THE PENTAGON WASHINGTON DC 20301-7100	3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DPTY CG FOR RDE HQ US ARMY MATERIEL CMD AMCRD MG CALDWELL 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001		<u>ABERDEEN PROVING GROUND</u>
1	INST FOR ADVNCED TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797	4	DIR USARL AMSRL CI LP (305)
1	DARPA B KASPAR 3701 N FAIRFAX DR ARLINGTON VA 22203-1714		
1	NAVAL SURFACE WARFARE CTR CODE B07 J PENNELLA 17320 DAHLGREN RD BLDG 1470 RM 1101 DAHLGREN VA 22448-5100		
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI MAJ M D PHILLIPS THAYER HALL WEST POINT NY 10996-1786		

NO. OF  
COPIES ORGANIZATION

8 DIRECTOR  
US ARMY RESEARCH LAB  
AMSRL IS  
J GRILLS  
AMSRL CI A  
R ROSEN (2 CPS)  
AMSRL CS  
COL K LOGAN  
AMSRL CI E  
W JERNIGAN  
W MCCOY  
B SCHALLHORN  
AMSRL SE S  
A SINDORIS  
2800 POWDER MILL RD  
ADELPHI MD 20783-1145

1 DIRECTOR  
US ARMY RESEARCH LAB  
AMSRL EA  
P STAY  
WSMR NM 88002-5501

ABERDEEN PROVING GROUND

14 DIR USARL  
AMSRL CI  
N RADHAKRISHNAN  
AMSRL CI E  
D ULERİY (10 CPS)  
AMSRL HR SF  
B AMREIN  
AMSRL WM IM  
R MCGEE  
AMSRL CS AP  
D ORE

# REPORT DOCUMENTATION PAGE

*Form Approved  
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>			<b>2. REPORT DATE</b> August 1999	<b>3. REPORT TYPE AND DATES COVERED</b> September 1997 - January 1998
<b>4. TITLE AND SUBTITLE</b>  Contract Req Software Requirements Analysis Version 1: A C-BASS Component			<b>5. FUNDING NUMBERS</b>  AC9UA4EC	
<b>6. AUTHOR(S)</b>  Denis McGurin				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  U.S. Army Research Laboratory ATTN: AMSRL-CI-E Aberdeen Proving Ground, MD 21005-5067			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  ARL-MR-458	
<b>9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)</b>			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b>				
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Approved for public release; distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (Maximum 200 words)</b>  This document contains the software requirements analysis for a prototype of Contract Request Version 1.0 (Contract Req). As a component of the Corporate Business Application Software System (C-BASS), this application automates the preparation of the requester's Procurement Data Package (PDP). The document follows the process of structured analysis, or step-wise refinement of requirements, as applied to the development of Contract Req. The "environmental model" includes a high-level system description, followed by a context diagram and a list of events to which the system must respond. The "behavioral model" includes a data flow diagram (DFD) for each of the five Contract Req subsystems. From this representation, the basic functional specifications are derived and represented in structured English (or program design language). The final segment of the document includes a data dictionary.				
<b>14. SUBJECT TERMS</b>  Contract preparation, formal contracts, user requirements, structured analysis, data flow diagrams, data dictionary			<b>15. NUMBER OF PAGES</b> 39	<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b> UNCLASSIFIED	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> UNCLASSIFIED	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> UNCLASSIFIED	<b>20. LIMITATION OF ABSTRACT</b> UL	

**INTENTIONALLY LEFT BLANK.**

## USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-MR-458 (McGurin) Date of Report August 1999

2. Date Report Received \_\_\_\_\_

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)  
\_\_\_\_\_  
\_\_\_\_\_

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.)  
\_\_\_\_\_  
\_\_\_\_\_

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate.  
\_\_\_\_\_  
\_\_\_\_\_

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)  
\_\_\_\_\_  
\_\_\_\_\_

Organization

CURRENT  
ADDRESS

Name

E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

Organization

OLD  
ADDRESS

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)  
**(DO NOT STAPLE)**

---

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

A vertical stack of seven thick horizontal bars, likely representing a postage indicia or a placeholder for one.

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR  
US ARMY RESEARCH LABORATORY  
ATTN AMSRL CIE  
ABERDEEN PROVING GROUND MD 21005-5067

---